Multimed Tools Appl (2010) 47:433–460
DOI 10.1007/s11042-009-0332-6

Music emotion classification and context-based music recommendation

Byeong-jun Han · Seungmin Rho · Sanghoon Jun · Eenjun Hwang

Published online: 5 August 2009

Abstract Context-based music recommendation is one of rapidly emerging applications in the advent of ubiquitous era and requires multidisciplinary efforts including low level feature extraction and music classification, human emotion description and prediction, ontology-based representation and recommendation, and the establishment of connections among them. In this paper, we contributed in three distinctive ways to take into account the idea of context awareness in the music recommendation field. Firstly, we propose a novel emotion state transition model (ESTM) to model human emotional states and their transitions by music. ESTM acts like a bridge between user situation information along with his/her emotion and low-level music features. With ESTM, we can recommend the most appropriate music to the user for transiting to the desired emotional state. Secondly, we present context-based music recommendation (COMUS) ontology for modeling user's musical preferences and context, and for supporting reasoning about the user's desired emotion and preferences. The COMUS is music-dedicated ontology in OWL constructed by incorporating domain-specific classes for music recommendation into the Music Ontology, which includes situation, mood, and musical features. Thirdly, for mapping low-level features to ESTM, we collected various high-dimensional music feature data and applied nonnegative matrix factorization (NMF) for their dimension reduction. We also used support vector machine (SVM) as emotional state transition classifier. We constructed a prototype music recommendation system based on these features and carried out various experiments to measure its performance. We report some of the experimental results.

B.-j. Han · S. Jun · E. Hwang (✉)
School of Electrical Engineering, Korea University, Seoul, Korea
e-mail: ehwang04@korea.ac.kr

B.-j. Han
e-mail: hbj1147@korea.ac.kr

S. Jun
e-mail: ysbhjun@korea.ac.kr

S. Rho
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: smrho@andrew.cmu.edu

Keywords Emotion state transition model · Music information retrieval · Mood · Emotion · Classification · Recommendation

1 Introduction

With recent advances in the field of music information retrieval, we face a new possibility that music can be automatically analyzed and understandable by the computer to some semantic level. Due to the diversity and richness of music content, many researchers [9, 15, 25] have been pursuing a multitude of research topics in this field, ranging from computer science, digital signal processing, mathematics, and statistics applied to musicology. Recent issues in music information retrieval include automatic audio genre/mood classification, music similarity computation, audio artist identification, audio-to-score alignment, query-by-singing/humming, multiple F0 estimation and tracking, and so on. One of the feasible applications is to provide content-based music recommendation. If we take one step further and utilize the context information, we can achieve more intelligent context-based music recommendation. For remarkable achievement in context-based music recommendation system, it often needs multidisciplinary efforts such as emotion description, emotion detection/recognition, low-level feature-based classification, and inference-based recommendation.

An emotion descriptor has been useful and effective in describing music taxonomy. An assumption for emotion representation is that emotion can be considered as a set of continuous quantities and mapped into a set of real numbers. As a pioneering effort to describe human emotions, Russel [27] proposed a circumflex model where each affect is displayed over two bipolar dimensions. Those two dimensions are pleasant-unpleasant and arousal-sleep. Thus, each affect word can be defined as some combination of pleasure and arousal components. Later, Thayer [31] adapted Russel's model to music. Thayer's model has "arousal" and "valence" as its two main dimensions. In this model, emotion terms were described as silent to energetic along the arousal dimension, and negative to positive along the valence dimension. With Thayer's model, the two-dimensional emotion plane can be divided into four quadrants with eleven emotion adjectives placed over them as shown in Fig. 1.

On the other hand, Xiang *et al.* [32] proposed a "mental state transition network" for describing emotion transitions of human beings. In the network, mental states consist of happy, sad, anger, disgust, fear, surprise, and serene. Every transition between two states is calculated from test data, and represented by some probability. However, they didn't consider other emotions such as nervous and excited.

Automatic emotion detection and recognition in speech and music is growing rapidly with the technological advances of digital signal processing and various effective feature extraction methods. Emotion detection/recognition can play an important role in many other potential applications such as music entertainment and human-computer interaction systems. One of the first researches on emotion detection in music is presented by Feng *et al.* [7, 33]. They implemented on the viewpoint of Computational Media Aesthetics (CMA) by analyzing two dimensions of tempo and articulation which are mapped into four categories of moods: happiness, anger, sadness and fear. This categorization is based on Juslin's theory [13] which describes the utilization of acoustic cues such as tempo and articulation in communication of emotions by performers and audiences: tempos were either fast or slow while articulations were either staccato or legato [33].

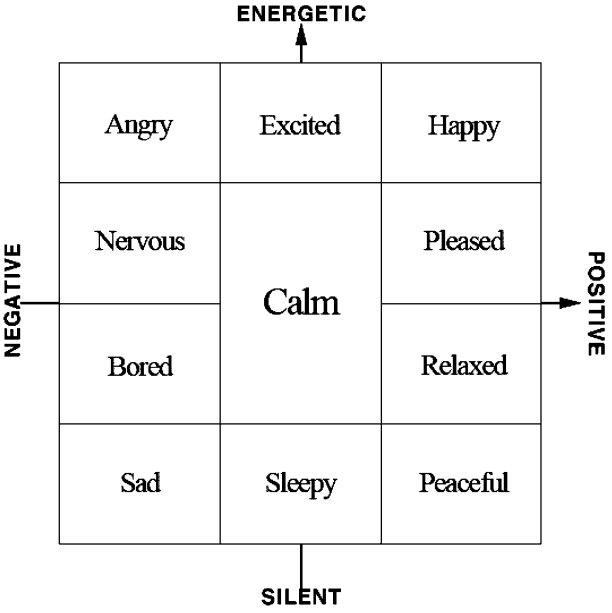


Fig. 1 Two-dimensional emotion representation in Thayer's model

For low-level feature-based music classification, traditional approaches have focused only on low-level features such as energy, zero crossing rate (ZCR), audio spectrum, and wavelet. A main deficiency of this approach is that these approaches relied on the physical context of contents only. More recent works have tried diverse human emotion descriptors. For example, in their work on automatic mood detection and tracking by Lu *et al.* [19], the authors classified various features into three categories: intensity, timber, and rhythm. All the moods were mapped into the Thayer's two-dimensional space [31]. Also, they used the Gaussian mixture model (GMM) as a classifier. To track music mood, they proposed a mood boundary detection method based on an adaptive threshold. Holzapfel *et al.* [10] also proposed a nonnegative matrix factorization (NMF)-based classification, wherein they calculated a sub-spectrogram as spectral bases of an audio signal and factorized them using NMF. They used GMM as classifier and expectation-maximization (EM) as a training algorithm. One of their main objectives was to detect static descriptors such as mood and genre. Lu *et al.* [19] tried to track mood for mood transitions in a song. To the best of our knowledge, there has been no effort thus far to measure and utilize the effect of music for human emotion state transition.

For more effective music recommendation, we need semantic information on the context as well as music. For example, assume that a person is very sad for some reason. Depending on his/her personality, he/she may want to listen to music that may cheer him/her up, or music that can make him/her calm down. Some researchers (e.g., Oscar *et al.* [21, 22]) observed and tried to bridge the semantic difference between low-level features and high-level concepts, which is known as a semantic gap. With low-level feature analysis only, we will experience many difficulties in identifying the semantics of musical content. Similarly, it is difficult to correlate high-level features and the semantics of music. For instance, a user's profile, which includes educational background, age, gender, and musical taste, is one possible high-level feature. Semantic web technology is considered to be one promising method to bridge this semantic gap.

In this paper, we address these problems in the domain of music recommendation by combining content-based music retrieval, music ontology, and domain-specific ontologies such as mood and situation. Specifically, based on the basic concepts from the upper ontologies which can be found in previous ontology-related projects such as Music Ontology [34, 35], Kanzaki taxonomy [37], and MusicBrainz [38], we define more specific domain-oriented ontologies. Music Ontology is an effort led by ZitGist LLC and the Centre for Digital Music to express music-related information on the semantic web. It attempts to express all the relations between musical information to help people to find anything about music and musicians.

1.1 Our contributions

In this paper, we propose three major contributions for context-aware music recommendation. First, we propose an emotion state transition model (ESTM) for representing complex human emotions and their transitions by music, and construct an emotion transition matrix (ETM) as our metric. Second, we propose an ontology called COMUS for representing various user situations, moods, and profile descriptors and reasoning desired emotion based on user's current emotion, situation and events occurred. COMUS is a combination of Music Ontology and our domain-specific ontologies such as mood and situation. Finally, we propose an emotion-transition classification for mapping between emotion-transition and low-level features of music. We used various low-level spectral and statistical features based on new framing to reflect the temporal context of music. To reduce the high dimensionality of low-level music features, we applied NMF, which recursively reduces and calculates the eigenvectors. We carried out various experiments in order to show the performance of our classification method.

The remainder of this paper is organized as follows. Section 2 describes our emotion state transition model (ESTM) and its mathematical representation. Section 3 represents our expanded music ontologies and demonstrates their usage using some scenarios. Section 4 shows the analysis of low-level music features and the overall training procedure. Section 5 describes the details of our implementation and some of the experimental results. Finally, Section 6 concludes this paper.

2 Emotion model and ESTM

In this section, we describe ESTM. This model is used to find and recommend to the user the most effective music for transition to the desired emotion. For this purpose, we first measure the emotion state transition effect of music in the database. Human emotion is very complicated and the resulting emotion states of human beings after listening to some type of music can be manifold and diverse. To handle this, we first try to represent complicated human emotions into a formal framework. Secondly, we describe the emotion state transition matrix which shows the degree of contribution to the transition for each pair of emotion states.

2.1 Concept of emotion state transition

Most previous approaches in music information retrieval have focused on the static features of music such as genre or mood, and they were quite successful in classifying music into some taxonomy. However, they are too strict to express certain diversity. For

example, in the case of genre, current music descriptors cannot describe some fusion genre. Another example is Shibuya-kei, a new genre which originated in Japan. It contains some features of electronica, lounge, bossa nova, and so on. However, we cannot put Shibuya-kei in a specific genre, because it is determined not from musical features but from the place of origin. Many people are already accustomed to fusion genres, which fuse two or more musical genres. For that reason, some people do not accept tailoring their musical taste into some classical music taxonomy. On the other hand, mood tracking by Lu *et al.* [19] demonstrated that music may no longer be described by one static mood. "Bohemian Rhapsody" (Queen) is one such example, where diverse mood variations occur. Also, traditional music descriptors cannot express the following situation efficiently: when a "sleepy" user wants some music to shake his/her mood up, previous approaches would recommend an "exciting" music. It might be quite effective for typical situations, but in some specific situation, this might not be a good solution. For example, if it is late night or the user is in the library, then 'exciting' music might cause inconvenience to the people nearby. In this case, soft or easy listening music might be a better solution to change moods.

In the paper, we concentrate on the 'transition of emotion state by music' which can be referred to as the 'effect of music.' This is deeply related to applications such as musical therapy, music recommendation, and context-based audio jukebox.

2.2 Complex emotion state

Literature on psychology provides diverse definitions on the human emotion [14]. So far, various efforts have been done to represent human emotion. For instance, Russell [27] and later Thayer [31] proposed two dimensional models based on the limited number of cognitive components such as arousal and valence. Component process theory is another approach to model human emotion. This theory [29] assumes that human emotion is composed of fundamental cognitive components. For example, angry is a combination of multiple emotions in mathematical representation. Since human being is accustomed to expressing his/her emotion using emotion adjectives such as happy, sad and angry, it is a challenging problem to define the cognitive components and their combination for each human emotion.

As an evidence, Scherer [28] presented a facial expression experiment where angry can be represented by four or more cognitive components and can be shared with other emotions such as discouragement, resolution and frightened [14].

On the other hand, many psychologists distinguish feeling from emotion. Feeling refers to the subjective experience of the emotion. Thus, it refers to physical experience in a narrow meaning and can be described clearly. Another approach to describing emotions is to use adjectives appropriate for an emotion of the human mind or event of interest. However, with the fact that emotion can occur unconsciously [29], it is very difficult to describe the emotion state of a human being at a moment using a word. For instance, a person might be happy as well as sad, or be both excited and calm at the same time. In many cases, people experience difficulty in describing their own emotion using a small number of emotional adjectives.

In our approach, we assume that human beings can experience different emotions at the same time, that each emotion at a moment can contribute to the transition to next emotion, and that the emotion transition occurs unconsciously. Also based on the component process theory, we expressed human emotion by the combination of emotion adjectives and their strength.

2.3 Mathematical representation

With the mathematical representation of emotion state transition, we can measure and manipulate the effect of music. For that purpose, we describe the emotion state of human beings using N adjectives and their strength values. Thus, for a set of emotion states $\mathbf{E}=\{e_1, e_2,\ldots, e_N\}$, where e_i represents an emotion such as 'happiness' and 'sadness,' we can describe the emotion state of a human being at a specific moment as follows:

$$\mathbf{ES}(t) = [es_1(t)\, es_2(t) \cdots es_n(t)]$$

where **ES** is the emotion state vector, and es_k is a nonnegative real value at any specific time t indicating the emotion strength. For example, if es_1=0.4, es_2=0.1, and all the other elements are zero, then the contributions of emotion es_1 and es_2 are 40% and 10%, respectively, and other emotions have 50% contribution. By combining emotion state vectors of initial and final moments, we can construct an emotion state transition matrix as follows:

$$\begin{aligned}
&\mathbf{ETM}\left(t_{init}, t_{final}\right)\\
&= \mathbf{ES}(t_{init})^{\mathbf{T}} \cdot \left(\mathbf{ES}\left(t_{final}\right) - \mathbf{ES}(t_{init})\right)\\
&= \left\{p_{n,m} \middle| p_{n,m} = es_n(t_{init}) \cdot \left(es_n\left(t_{final}\right) - es_m(t_{init})\right)\right\}\\
&= \begin{bmatrix}
p_{1,1} & p_{1,2} & \cdots & p_{1,m} & \cdots & p_{1,N}\\
p_{2,1} & \ddots & & & & \vdots\\
\vdots & & \ddots & & & \vdots\\
p_{n,1} & & & p_{n,m} & & \vdots\\
\vdots & & & & \ddots & \vdots\\
p_{N,1} & \cdots & \cdots & \cdots & \cdots & p_{N,N}
\end{bmatrix}
\end{aligned}$$

where t_{init} and t_{final} are initial and final moments, respectively. $p_{n,m}$ indicates the (n,m)-th element of the emotion state transition matrix. For example, we assume that we have three emotion states and one source that causes emotion transition, and the **ES**s of the initial and final state are {0.4, 0.3, 0.2} and {0.7, 0.3, 0}, respectively. Then, the difference of initial and final state is {0.3, 0, −0.2} and their transition matrix is [0.12 0.0 −0.08; 0.09 0.0 −0.06; 0.06 0.0 −0.04]. We can observe three properties in this example. First, the source has a positive effect on the emotion transition of the first emotion, because the values in the first column are all nonnegative values. Second, the source does not have an effect on the emotion transition to the second emotion, which can be observed from the second column whose values are all zero. Third, the source has a negative effect on the emotion transition to the third emotion, because the values on the third column are all negative values.

The difference between our definition and the mental state transition network [32] is that the latter did not consider negative effects from the source, but rather a probabilistic representation between mental states. Thus, it fails to explain negative effect on the emotion transition.

3 Ontology model

As previously mentioned, in order to provide music recommendation service intelligently, we need a set of common ontologies for knowledge sharing and reasoning. We have developed music and its related ontologies in the music recommendation domain. We use the W3C recommendation ontology language Web Ontology Language (OWL) to represent

ontology. OWL helps represent a domain by defining classes and properties of those classes, to define individuals and their properties, and to reason about these classes and individuals.

The OWL language is derived from the DAML+OIL language, and both are layered on top of the standard RDF(S) triple data model (i.e., subject, predicate, and object). Based on the basic concepts and relations from previous work—the Music Ontology [39], we expand this work to include additional features such as musical feature, genre, instrument taxonomy, mood, and situation. We serialize these ontologies by OWL so that we can retrieve information using the SPARQL query language.

3.1 COMUS ontology

The COMUS ontology provides an upper Music Ontology that captures concepts about the general properties of music such as title, artists and genre and also provides extensibility for adding domain-specific ontologies, such as Music Feature, Mood and Situation, in a hierarchical manner. Music Ontology [34, 35] is an effort led by ZitGist LLC and the Centre for Digital Music to express music-related information on the semantic web. It attempts to express all the relations between musical information to help people to find anything about music and musicians.

The COMUS ontology consists of 18 classes and 32 property definitions. Figure 2 shows a representation of some of key COMUS ontology definitions. This ontology describes music related-information about relationships and attributes that are associated with people, genre, mood (e.g., angry, happy), location (e.g., office, street), time (e.g., morning, winter), and events (e.g., driving, working) in daily life.

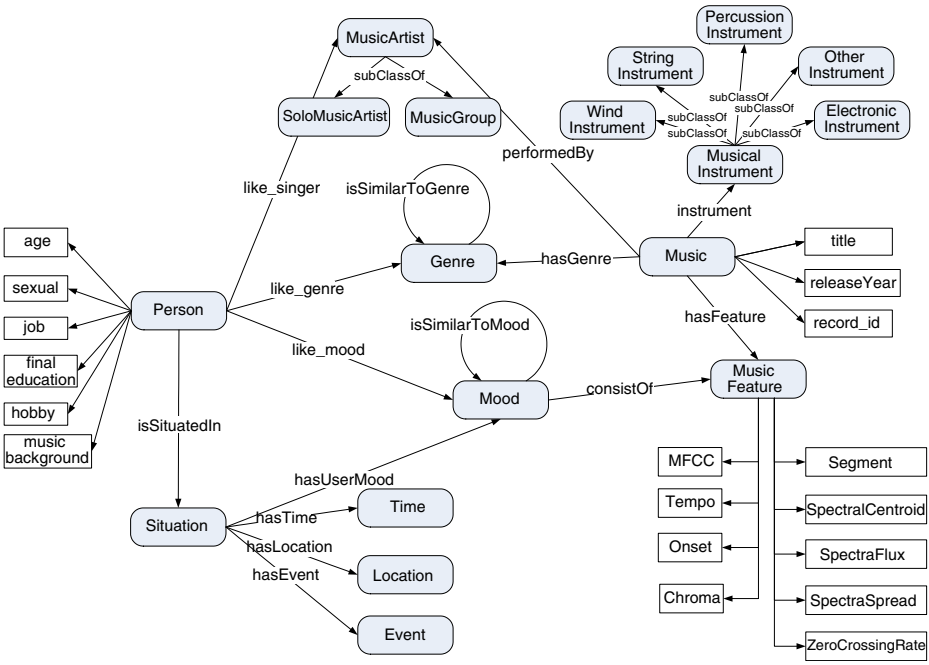


Fig. 2 Diagram of our COMUS ontology. Each rounded rectangle with solid arrow represents an OWL class

Figure 3 shows part of the COMUS ontology in XML syntax. The key top-level elements of the ontology consist of classes and properties that describe Person, Situation, Mood, Genre and Music classes. In the following, we will briefly describe each of the classes and show a few SPARQL query examples for the scenario which we will discuss in the next section.

Person The "Person" class defines generic properties of a person such as name, age, gender, hobby, socioeconomic background (e.g., job, final education) and music related properties for music recommendation such as musical education, favorite music, genre, and singer.

Situation The "Situation" class defines a person's situation in terms of conditions and circumstances, which are very important clues to effective music recommendation. Hence, this class describes the user's situational contexts such as the whereabouts of the user (Location), what happens to the user (Event), and so on.

The following questions describe how this ontology might be used for evaluating the user's situational contexts and recommending appropriate music.

Q1: What kinds of music does he/she listen to when he/she feels gloomy in her bed at night?
Q2: What kinds of music does he/she listen to when he/she takes a walk to enjoy a peaceful afternoon?
Q3: What kinds of music does he/she want to listen to when he/she needs to keep his/her current emotion or change into different emotion?

```
<owl:Class rdf:about="#Feature">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        MFCC, Chroma, Average Energy, … </rdfs:comment>
   <rdfs:subClassOf rdf:resource="#MusicExtension"/>
</owl:Class>
<owl:Class rdf:about="#MFCC"><rdfs:subClassOf rdf:resource="#Feature"/></owl:Class>
<owl:Class rdf:ID="Mood"><rdfs:subClassOf rdf:resource="#MusicExtension"/></owl:Class>

<owl:ObjectProperty rdf:ID="hasLocation">
   <rdfs:domain rdf:resource="#Situation"/>
   <rdfs:range rdf:resource="#Location"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasUserMood">
   <rdfs:domain rdf:resource="#Situation"/>
   <rdfs:range rdf:resource="#Mood"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="like_mood">
   <rdfs:range rdf:resource="#Mood"/>
   <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="like_genre">
   <rdfs:domain rdf:resource="#Person"/>
   <rdfs:range rdf:resource="#Genre"/>
</owl:ObjectProperty>
```

Fig. 3 Part of our proposed ontology in XML syntax. The complete ontology is available at [40]

Mood The "Mood" class defines the state of one's mind or emotion. Each mood has a set of similar moods. For example, "aggressive" has similar moods such as "hostile, angry, energetic, fiery, rebellious, reckless, menacing, provocative, outrageous, and volatile." We collected the mood terms which are obtained from All Music Guide [41]. Then, we classified these mood sets into 11 distinctive mood adjectives based on Thayer's emotion model.

Genre The "Genre" class defines the category of music. There have been many studies in music genre classification. In the music industry, music taxonomies are usually made up of four levels: (i) Top level consists of global musical categories such as Classical, Jazz, and Rock; (ii) Second level consists of specific sub-categories (e.g., "Hard Rock" within "Rock"); (iii) Third level usually gives an alphabetical ordering of artists (e.g., "Queen"); (iv) Fourth level consists of tracks from the artist's album [23].

There exist several popular online systems such as All Music Guide, MusicBrainz [41], and Moodlogic [44] for annotating popular music genre and emotion. We create our own genre taxonomy based on the All Music Guide along with a second level of industry taxonomy.

Music The "Music" class defines general properties of music such as title, released year, artists, genre, and musical features (e.g., MFCC, Tempo, Onset, Chroma, Segment, Spectra Centroid, Spectra Flux, Spectra Spread, and Zero Crossing Rate).

3.2 Scenario: situation-based music recommendation

In this section, we introduce a typical scenario that demonstrates how COMUS ontology can be used to support ontology reasoning for recommending appropriate music to users.

Example Scenario

Tom is a technical consultant and he is 42 years old. His favorite band is "Chicago," and he likes pop and hard rock-style music. His hobby is playing baseball. He is a very positive person and likes bright, soft, and sweet music. When he feels sad or gloomy, he usually listens to the music that might help cheer him up.

The date is 23 June, 2008. Tom woke up late on Monday morning and he is still very tired, having worked late last night. He has to go to work early to prepare for a presentation at a meeting. Therefore, he requests the music recommendation system to look up some hard and fast beat music, and the system plays music including "Welcome to the Jungle" (Guns N' Roses), "Heartbreaker" (Nirvana) and "I Remember You"(Skid Row). This kind of music could help him to hurry up and go to work on time.

On the way to work, he was stuck in a traffic jam which started making him nervous. In order to calm down his mood, he requests the system to recommend some music for the situation and the system plays music including "Top of the world" (Carpenters), "Lost in love" (Air Supply), and "The boxer" (Simon & Garfunkel).

This scenario assumes that Tom set his musical preferences, such as singer, genre, and mood, to filter out the query result automatically. For example, considering the wake-up-late scenario, the situation information described in Fig. 4 is analyzed and sent to the recommendation system. Using this information, the system will evaluate Tom's situational

```
<Situation rdf:ID="S1_IntheMorningWakeupLately">
    <hasEvent> <UserEvent rdf:ID="WakingUp"/> </hasEvent>
    <hasTime> <Time rdf:ID="Morning"/> </hasTime>
    <hasSituationName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Wake Up Late</hasSituationName>
    <hasUserMood rdf:resource="#Plaintive"/>
    <hasUserMood rdf:resource="#Bright"/>
    <hasUserMood rdf:resource="#Cheerful"/>
    <hasUserMood rdf:resource="#Gloomy"/>
    <hasLocation> <Location rdf:ID="BedRoom"/> </hasLocation>
  </Situation>
```

Fig. 4 Example situation ontology in XML syntax

context and his favorite mood from the user profile information. From this information, the system recommends music which best fits Tom's interests and current situation.

Based on the situation described in Fig. 4 and Tom's favorite mood from his profile information, we might formulate a user query as in Fig. 5(a) and its result shown in Fig. 5(b).

```
PREFIX mol: <http://mil.korea.ac.kr/ ontology/0.1/musicontology.owl#>
SELECT DISTINCT ?Song ?UserMood
FROM <http://mil.korea.ac.kr/ontology/0.1/musicontology.owl>
WHERE
{
?Person mol:hasName "Tom";
mol:likeMood ?UserMood.
?Situation mol:hasSituationName "Wake Up Late";
mol:hasUserMood ?UserMood.
?UserMood mol:hasSimilarMood ?SimiliarMood.
{?Song mol:hasMoods ?UserMood} UNION {?Song mol:hasMoods ?SimiliarMood}
}
ORDER BY ?Song
```

(a)

Results

Song	UserMood
Billie_Jean	Bright
Dancing_Queen	Bright
Dont_Stop_Me_Now	Bright
Fernando	Bright
Hey_Jude	Bright
I_Want_To_Hold_Your_Hand	Bright
Lay_Your_Hands_On_Me	Bright
Mamma_Mia	Bright
One	Bright
Somebody_To_Love	Bright
Top_Of_The_World	Bright
Waterloo	Bright
We_Are_The_Champions	Bright
Yesterday_once_more	Bright

(b)

Fig. 5 **a** Sample SPARQL query in the "Wake up late" situation **b** The query result

4 Classification

In this section, we describe how to associate low-level features of music with emotion state transitions of human beings. To extract low level features of music, we first perform beat tracking-based framing on the music signal and extract various spectral and statistical features from each frame. Since the amount of feature data we extracted was extremely large, which is known as the *curse of dimension*, we performed dimensionality reduction using NMF in the temporal direction. Finally, feature vectors are trained using the support vector machine (SVM) for mapping into the emotion state transition matrix.

4.1 Beat tracking-based framing

Usually, to extract features from a music signal, several preprocessing steps and framing should be performed including noise reduction, amplitude or volume normalization, and so on. Framing decomposes a music signal into smaller equally sized temporal units. It is known that a smaller frame size can lead to better time resolution in spectral analysis, but degraded frequency resolution [15]. In the typical music classification approach, the frame size is larger than 100 ms to ensure reasonable frequency resolution.

Whereas equal size framing might be efficient in statistical analysis, it is difficult to determine the meaningful features of each frame. With meaningful features of the frames, music analysis could be done more easily and efficiently. In the case of music beats, they provide useful information for detecting meaningful units of music.

For detecting musical beats, we used the beat tracking algorithm of Ellis and Poliner [6], which extracts both beat and overall tempo from a song. It detects beats by clipping the first differentials between phase-domain spectra in Fourier analysis. Even though this approach gives high accuracy, its result might include some errors. We assumed that songs have beats in the range from 50 beats per minute (BPM) to 300 BPM. We decided this BPM range based on the BPM Database [42] which provides BPM statistics on the pop music clips worldwide. In the rare case, music might have a BPM beyond the range such as Swans' song and extreme music genre such as gabber and grindcore. Since our recommendation system excludes this extreme music, 50 to 300 BPM range is enough for our purpose. If we convert the range into reciprocal form, its beat length becomes between 200 and 1200 ms. Figure 6 shows the difference between traditional framing and the beat tracking-based framing we used.

4.2 Feature extraction

Some perceptual features are known to be represented by physical measurements such as intensity, brightness, timbre, and so on. We extracted a variety of features from frames and normalize them to compensate for differences in frame lengths. These features can be divided into two categories: temporal and spectral.

Since human perception of a signal sequence is affected by the auditory organ, a psychoacoustic model can be applied to the result of discrete Fourier transform. This can be represented by the following equation:

$$\hat{f}(k) = \sum_{k-1}^{N} x(t) \cdot w_{psy}(k) \cdot e^{\frac{-j2\pi kt}{N}}$$

where f refers to the discrete Fourier transform and x is source signal. Thus, $\underline{f}$ returns an N-sample Fourier spectrum, w_{psy} is the psychoacoustic function, and its length is the same

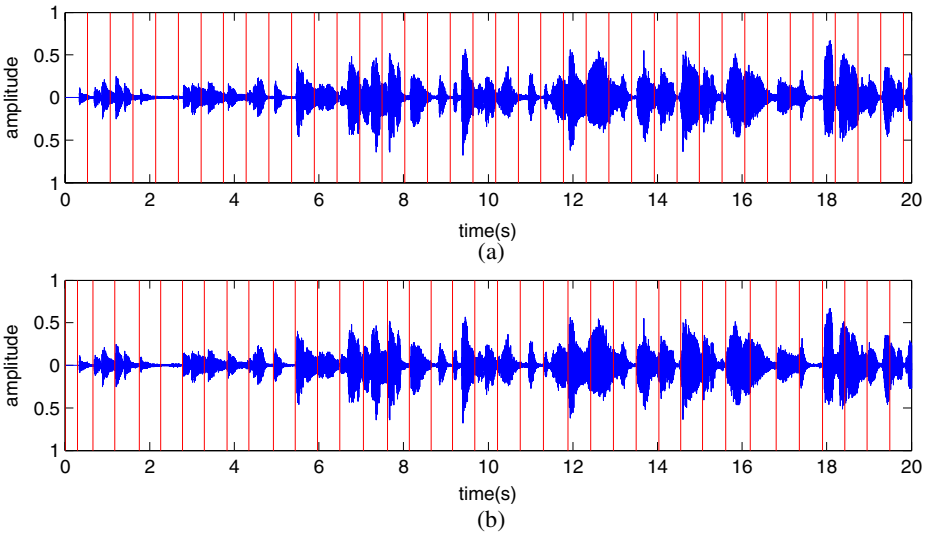


Fig. 6 Two framing methods: **a** traditional framing and **b** beat tracking-based framing

as f. We obtain a new magnitude of spectrum by the inner product of the magnitudes of f and w_{psy}.

4.2.1 Temporal features

Temporal features usually give the rhythmic property of music. In this paper, we considered various temporal features including intensity, variation of beat length, chord/scale variations, and so on.

Intensity The energy of the unit frame is usually referred to as the intensity of the frame, as in average energy (AE) or its spectral analysis representation [19], [15], [20]. Also, human beings can feel intensity of music perceptually [36]. In our previous work [12], we showed that the average and deviation of AE could express the emotion property of music efficiently. We extend the intensity representation in the temporal domain, and apply to it the psychoacoustic model of [36]. The reason we used the psychoacoustic model in this paper is that it gives better accuracy than the traditional intensity feature. A well-known psychoacoustic model which is also adopted in ISO 226:2003[11] was proposed by Robinson *et al.* [26]. According to ISO 226:2003, our intensity feature is defined by the following equation:

$$Intensity = \sum_{k=1}^{N} 4.2 + \frac{a_k\left(\hat{f}(k) - \mathrm{T}(k)\right)}{1 + b_k\left(\hat{f}_n(k) - \mathrm{T}(k)\right)}$$

, where T is a threshold function which represents a minimum perceptible intensity at frequency k, and both a_k and b_k are characteristic values at frequency k. T, a_k, and b_k are defined in [11]. Figure 7 shows a sample intensity feature.

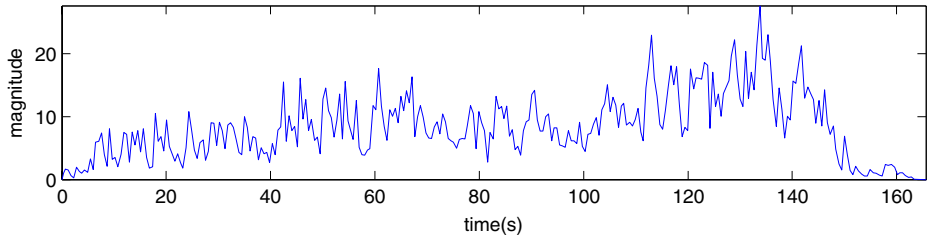


Fig. 7 Intensity feature extraction

Variation of beat length Generally, the global tempo of music gives a static property to rhythm. However, in many cases, music has several variations in rhythm. This diversity causes listeners to experience diverse emotions. Hence, variation of beat length in music can trigger a user's emotion transition from one state to another.

In order to represent variations in beat length, we first need to analyze beat tracking. Then, we construct vectors to represent the temporal sequence of beat lengths. By observing the difference of neighboring temporal vectors, we can compute the variation of beat lengths. This can be summarized by the following equation:

$$\mathrm{VBL}(n) = \frac{60}{GlobalTempo}\left(BeatLength(n+1) - BeatLength(n)\right)$$

In the preceding equation, *GlobalTempo* represents the global tempo obtained by a beat tracking or tempo estimation algorithm. *BeatLength*(n) is the duration of n-th frame. Figure 8 shows a sample VBL feature according to this equation.

Chord/scale variations Chord and scale variations are important to user perceptions. When the same or similar chords are repeated throughout the entire music, then the listener might become bored and reluctant to listen to it. Similarly, if the scale variation of music is dynamic, then the listener may feel stressed by it.

Chords and scales can be analyzed using chromagrams and spectrograms, respectively. A chromagram of each frame indicates where the notes are concentrated. Based on it, exact chords can be computed using key profile matrix [16]. Similarly, scales can be computed by grouping and comparing intensities of scales. After detecting chords and scales, we compute their differences for the current and next frames. Figure 9 shows a sample chord variation feature.

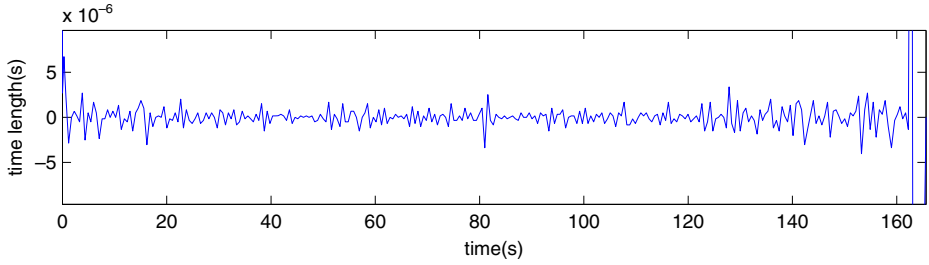


Fig. 8 Variation of beat lengths

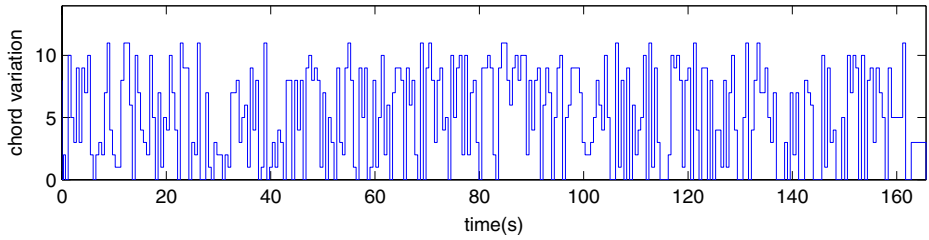


Fig. 9 Chord variation

4.2.2 Spectral features

The literature on acoustics shows that spectral features can be extracted from a wave sequence. Even though some of them are not enough to give independent meaning from other types of features, still they are very popular in MIR**.** In this paper, we used the following features for classification: spectral centroid (SC), spectral flatness (SF), spectral spread (SS), spectral flux (SFX), and mel-frequency cepstral coefficients (MFCC).

SC indicates brightness or sharpness of a sound and characterizes the centre of gravity of the spectra. Thus, for example, if the sound consists of many high frequency peaks, people might perceive that scale is very high, and even some people might feel noisy and uncomfortable. SF indicates how flat the spectrum of a signal sequence is. If SF is too high, then the sequence might be tonal. Otherwise, the sound might be noisy. Usually, human beings are very sensitive to noisy sound, and feel uncomfortable and even annoying. SS indicates average spread of the spectrum and is related to its own SC. If the value is high, then the spectral components are distributed widely, or even don't have any regularity in them. Otherwise, it indicates that some components are concentrated on a specific point. SFX represents local spectral change. People are sensitive to

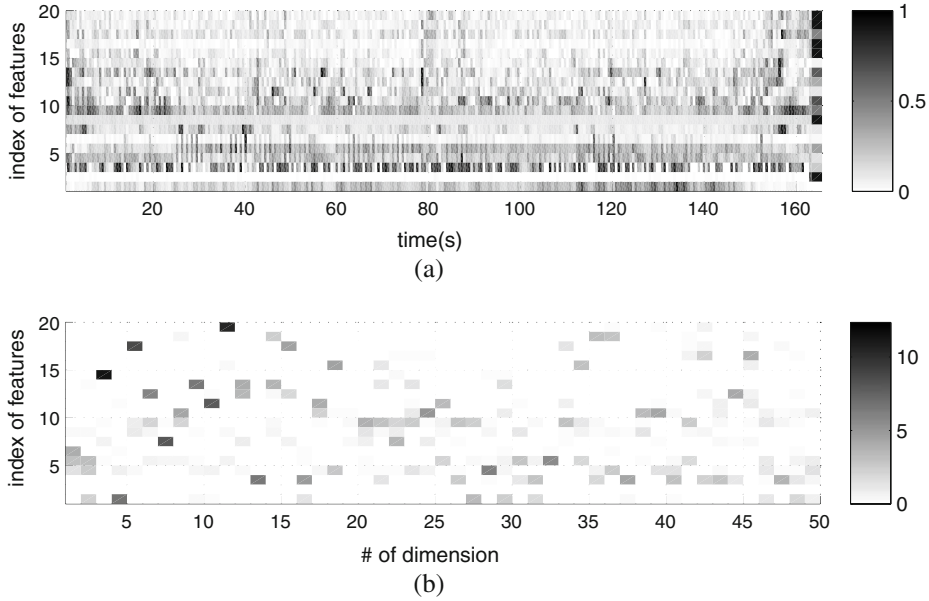


Fig. 10 Effect of dimensional reduction: **a** raw data and **b** after NMF

spectral changes. If the spectral changes are frequent, then people usually perceive diverse sounds, or even become uncomfortable for the sound. Otherwise, they feel serene. The detailed procedures to extract SC, SF, SS, and SFX are referred in [15] and [4].

MFCC represents the cepstral coefficients in mel scale. While human perceives the sound scale logarithmically, frequency-domain represents the scale in the linear space. Because of this mismatch, we use the mel-scale since it is represented in a logarithmic scale. The detailed procedure for computing MFCCs are described in [24].

4.3 Dimensionality reduction and training

To handle various features of music efficiently, the data are usually captured into high-dimensional feature vectors. It is not easy to train classifiers using these high-dimensional input data directly. Due to this, the dimensional reduction of these feature vectors is highly recommended. There are many methods for this purpose which show minimal loss of information.

In this paper, we used NMF [17], [18] to reduce the feature dimension of music. In order for NMF to work on our feature vectors, all elements of the feature vectors should be nonnegative. Also, elements of our features represent human perception. Thus, we removed the sign of elements in the feature vector and applied NMF. Figure 10 illustrates the effect of dimentional reduction when applying NMF. Detailed steps for factorizing nonnegative eigenvectors are represented in [18].

For classification, we used SVM. Since SVM classifies only one class at a time, we combined multiple SVMs to determine the effect of emotion transition [1]. More specifically, we used the results of dimensionality reduction as the training set and made label vectors for each song. After training SVM, we stored the result for verification. We use the marginal result to decide whether the song is effective on a specific emotion state transition.

5 Music recommendation system

Based on the ESTM, COMUS ontology and feature-based music classification, we implemented a prototype music recommendation system. In this section, we first present the overall system architecture with brief explanation for major components and then describe how to generate the ESTM from the SVM result.

5.1 Architecture

Our music recommendation scheme is based on user's current emotion, desired emotion, and the music-to-ESTM mapping. A typical procedure for music recommendation is as follows: when user inputs his/her personal information, musical preference and the favorite situation into the COMUS ontology, the system performs reasoning on his/her desired emotions. After that, database is queried to find out music that may trigger the state transition from current emotion to the desired emotion. Every song in the music database has its own emotion state transition matrix, which was generated from the classifier.

It is very time-consuming and cumbersome for the user to input his/her emotion and its transition to ESTM directly. COMUS ontology helps to fill out their gap using both user preference and situation information. Here, user preference includes both personal information and musical taste and situation information includes temporal and location-related data and event description.

Figure 11 represents the overall architecture of our prototype music recommendation system. Our system can be divided into two major parts: one part is in charge of ontology-based reasoning for proper music recommendation. The other part is in charge of low-level feature extraction and analysis, feature training, and mapping to ESTM.

In the ontology part, the user's desired emotion is deduced from the ontology based on the user situation, musical taste, user preference, etc. Our COMUS ontologies are described in OWL using Protégé editor [43]. Also, in order to express queries to our ontologies and process them for the recommendation, the Jena SPARQL engine is used. For example, when a user creates a profile containing his/her musical preference, it is not reasonable to expect him/her to specify all the required details into the profile. In that case, missing valuable information could be inferred from the partial information in the profile. As another example, the recommendation system could modify the playlists according to his/her current or desired emotion based on the users' profile (e.g., genre or artist preference) and listening habits (e.g., weather, time and location).

In the low-level feature analysis, various low level features such as temporal features (e.g., intensity, chord variations) and spectral features (e.g., spectral centroid, spread, spectral flatness, spectral flux, MFCC) are extracted from each frame. Next, NMF is used to reduce their high dimension. After training SVM using the ground-truth data, emotion state transition matrices are constructed from musical features of music clips.

5.2 Interfaces

We implemented a prototype music recommendation system using JSP and the Jena framework for the client and server sides, respectively. Our system provides various

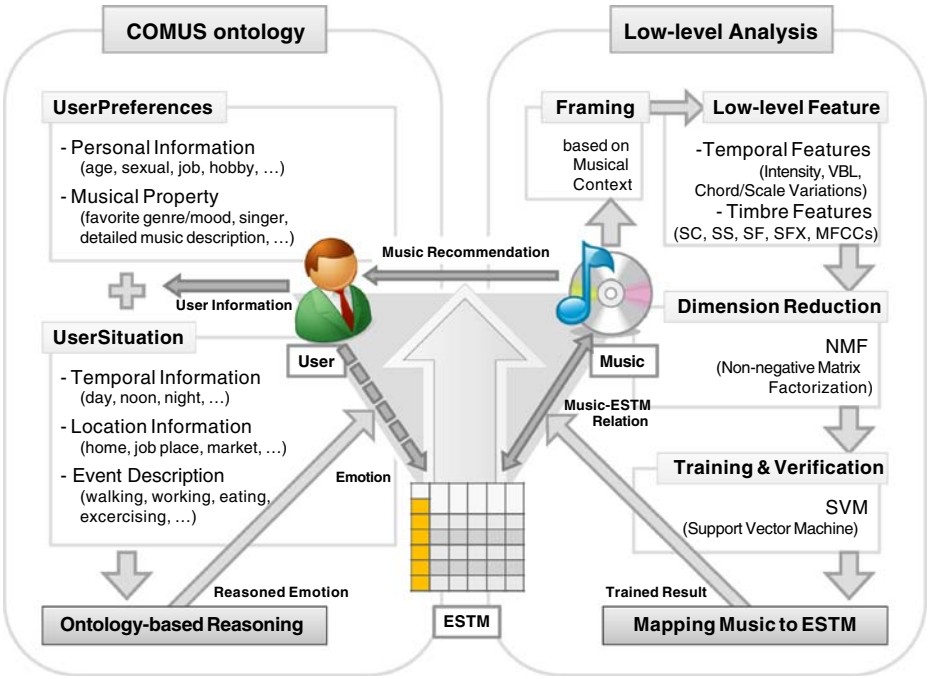
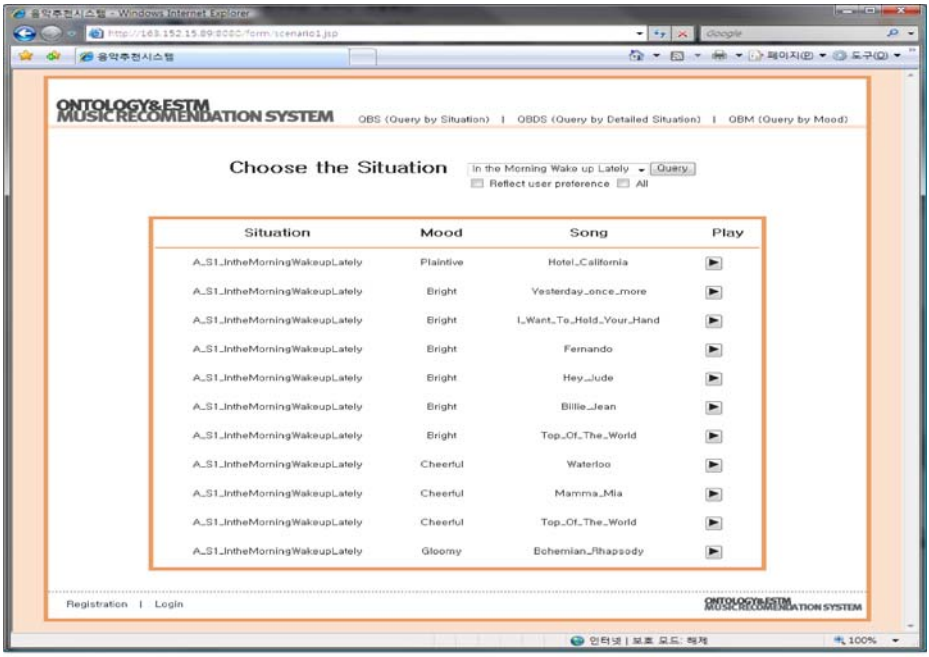
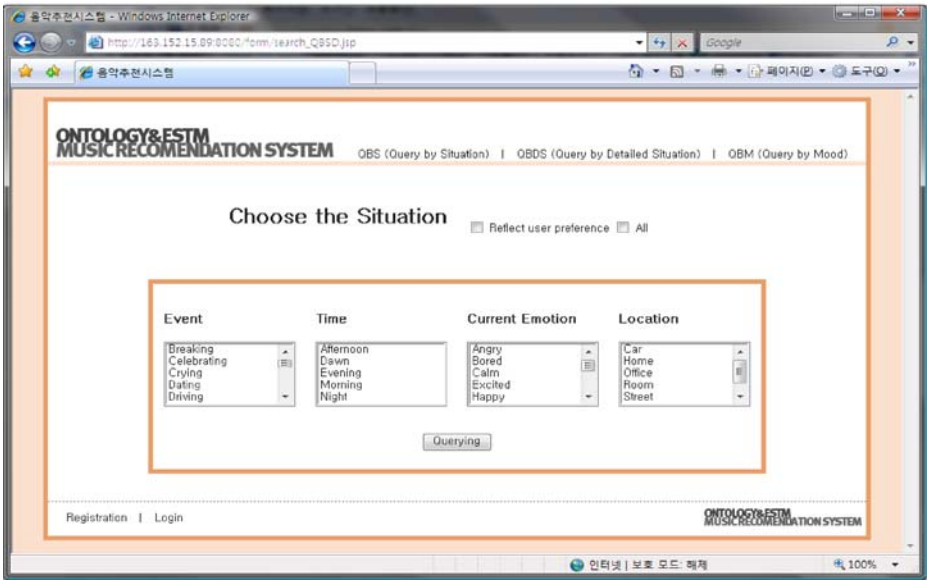


Fig. 11 Architecture of our music recommendation system



(a)



(b)

Fig. 12 Screenshots of query and result interface: **a** Query by situation (QBS) interface, and **b** Query by detailed situation (QBDS)

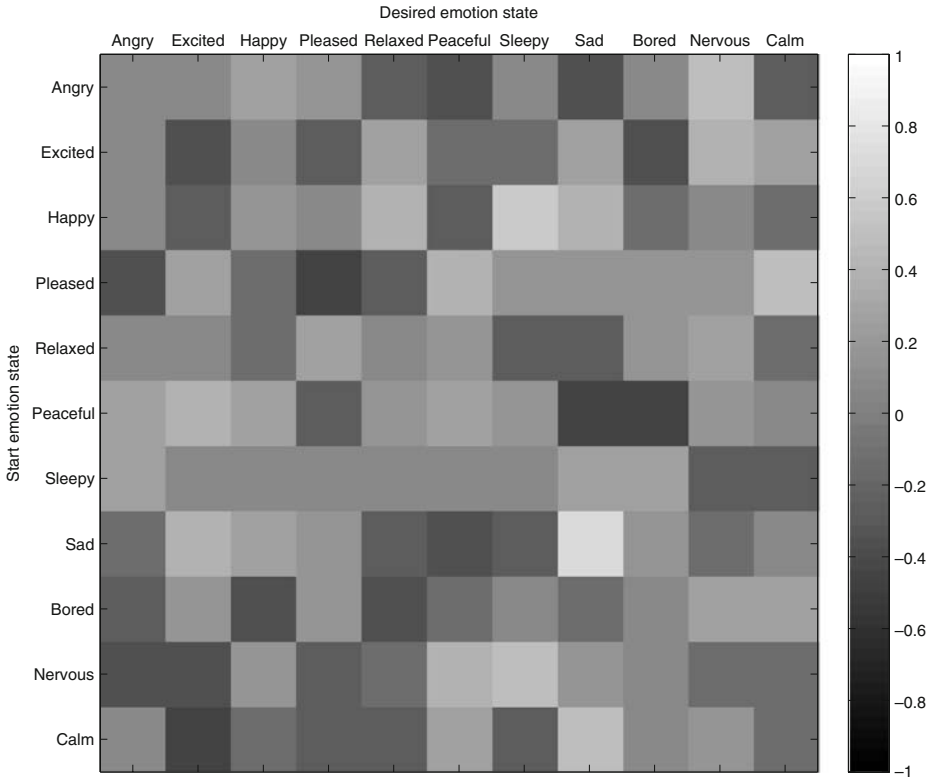


Fig. 13 Emotion state transition matrix for the song "A Wink of a Smile"

types of query interfaces to the users. The user can formulate queries using one of three different query interfaces: they are query by situation (QBS), query by detailed situation (QBDS), and query by mood (QBM). Figure 12 shows the result of the query described in Fig. 5(a). As shown in this figure, the system recommends a list of songs based on the user's favorite musical mood "bright," and its similar mood "cheerful."

5.3 Emotion state transition matrix

In this section, we describe how to generate ESTM from music features using the SVM classifier. Generating ESTM can be divided into two phases: training and classification. Since we use positive and negative probabilistic representation for ground-truth collection, each ESTM entry is represented by a numerical value in the range from -1(−100%) to 1 (100%). Some classifiers such as Neural Networks (NN) and SVM can also express the verification result into −1 through 1. We used SVM because of its performance and simplicity in the representation of verification result.

In the training phase, we first extracted music features and reduced their dimensionality using NMF. Next, we collected the ground-truth emotion state transitions from volunteers. Each ground truth value has a probabilistic representation such as 70% true (or 40% false) to indicate its positive (or negative) contribution. And then for each entry in the emotion state transition matrix, we trained its SVM using the

ground-truth emotion state transition matrices. For each input data, the trained SVM returns a value indicating its distance from the hyperplane. In particular, if the distance is bigger than the maximum margin which was generated during the training phase, the input data is marked as true (100%) or false (−100%). Otherwise, the distance is divided by the maximum margin to represent a probability in the range of −1 and 1, exclusively. An example of emotion state transition matrix is depicted in Fig. 13.

6 Experiments

In this section, we describe experiments to measure the performance of our system, and we show some of the results.

6.1 Mood mapping by ontology

According to [8], the development of ontology is motivated by scenarios that arise in applications. A motivating scenario provides a set of intuitively possible solutions to the problems in the scenario. The COMUS ontology is a collection of terms and definitions relevant to the motivating scenario of music recommendation that we described previously. Thus, in building the ontology, it is important to start by describing the basic concepts and one or more scenarios in the specific domain of interest. After building the ontology, the next step is to formulate competency questions. These are also based on the scenarios, and can be considered as expressiveness requirements that are in form of questions. Ontology must be able to represent these questions using its domain-related terminology and characterize their answers using axioms and definitions. Therefore, we asked participants to answer competency questions through an online questionnaire system.

In the experiment, we had about 30 participants. Some of them are musically trained, and others are not. Participants were asked to fill out the questionnaire to collect suitable terms and definitions about situation and mood. They were also requested to describe their own emotional state transitions such as current and desired emotions in the specific scenario. The description was based on one or more emotional adjectives such as happy, sad, angry, nervous, and excited–these were collected from the All Music Guide taxonomy [41]. Finally, the most frequently described adjectives were chosen to define the instances in the COMUS ontology.

After building the ontology, we performed an experiment to measure the level of user satisfaction using either our proposed COMUS ontology or the AMG Taxonomy in our system. The procedure for the experiment was as follows:

1) The experimenter explained the procedure and the purpose of the experiment, and demonstrated how to run our music recommendation system.

Table 1 User satisfaction for COMUS ontology and AMG taxonomy

	(unsatisfied) <———		(neutral) ———		(very satisfied) ———>
	1	2	3	4	5
AMG Taxonomy	1	3	19	5	2
COMUS Ontology	0	2	4	16	8

2) The participant described his profile (e.g., musical preferences) using web form interfaces such as buttons, textbox, checkbox, and selection list.
3) All participants were told to describe the situation, current emotion, and their desired emotion, or to select the predefined scenario using the query interfaces as described in Section 5.2.
4) Then, the system returned recommended songs based on the ontology reasoning and the participant's profile. The participant then judged which one was appropriate for their current emotion. Participants chose one selection on the 5 point rating scale (from 1=strongly unsatisfied to 5=strongly satisfied).
5) Finally, all the participants were asked to fill out a questionnaire.

As shown in Table 1, over 80% of the participants responded positively to the overall satisfaction of the system using ontology instead of AMG taxonomy. The results of the satisfaction ratings show that most of the users were satisfied with the query results recommended by the system.

With regard to the satisfaction of the participant's preferred emotional adjectives depicted in Table 2, positive adjectives (such as happy and excited) are found to be satisfactory to about 78% of the participants, whereas ambiguous adjectives such as nervous are found to be satisfactory to 43% of the participants (in the case of using COMUS ontology).

6.2 Dataset

To construct a fair dataset for music classification, we gathered ground-truth set and songs for ESTM and a music database, respectively. For the ground-truth set, we asked 30 volunteers to fill out emotion state transition effects through a web-based emotion state transition survey system. The survey was performed under various situations; that is, in a lecture room, outside, in a moving car, or on a train via wireless connection. Also, 120 songs were used in the survey. Each song was composed by a different artist group, sung by different singer, and had different lyrics.

Table 2 Participants' preferred emotional adjectives

	AMG Taxonomy					COMUS Ontology				
	1	2	3	4	5	1	2	3	4	5
angry	4	5	9	8	4	1	6	11	9	3
bored	9	12	6	2	1	1	3	6	11	9
calm	2	6	9	9	4	2	5	12	7	4
excited	3	4	6	8	9	0	3	6	8	13
happy	4	8	10	6	2	0	2	2	9	17
nervous	3	15	8	4	0	4	5	8	8	5
peaceful	3	7	8	6	6	0	3	5	14	8
pleased	6	7	13	3	1	0	1	5	9	15
relaxed	6	7	12	3	2	1	4	12	6	7
sad	4	8	16	2	0	0	1	14	11	4
sleepy	0	9	11	6	4	2	4	6	12	6

The procedure for collecting emotion state transitions from participants was as follows: first, participants described their current emotions in terms of relative percentages before listening to any songs. Second, participants listen to a randomly selected song and described their emotions in percentages. If participants wanted to carry out the survey continuously, then they repeated the steps.

In the experiment, we used 11 adjectives to describe user emotions: angry, excited, happy, pleased, relaxed, peaceful, sleepy, sad, bored, nervous, and calm. Thus, for example, a participant might describe his/her emotions as follows: 20% angry and 30% excited. We did not care about the size of their summation, because a user describes emotion transition by increasing or decreasing his/her emotions after listening to the song.

In the experiment, users are allowed to listen to same song several times. This is because listening to a song only once might not be enough to cause emotion transition. Hence, we counted how many times the user listened to a song to get a more accurate understanding of the effect of the song. For example, if a user listened to a song 3 times and input 30% in a specific emotion transition, then we divided the input value by 3.

Overall, participants input 32,573 evaluations on the emotion transitions for various songs.

6.3 Classification result

Our classification takes dimension-reduced music feature data and classifies it for emotion state transitions as input. We used LIBSVM v2.86 [3]. LIBSVM provides various types of SVMs and kernels. We selected both original SVM [2], [5] and one-class SVM [30] to compare classification performance.

Coefficients for SVM and kernels are very critical to performance. In our experiment, we tried several coefficients and kernels to empirically find the optimal classification setup. We also considered both the traditional method and the ν-fold cross-validation method in order to prevent the over-fitting problem. We tested ν-fold cross-validations using different ν values and obtained ν SVMs. Among these SVMs, we found the best SVM, which minimizes the overall error rates.

6.3.1 Performance evaluation

We evaluated the performance of SVMs in terms of three factors: overall accuracy, standard deviation (STD) of accuracy, and maximum accuracy.

First, overall accuracy describes how many correct outputs the classifier can predict among all the classes, and can be evaluated by the following equation:

$$Overall\ Accuracy = \frac{\sum_{s=1}^{N}\sum_{e=1}^{N} \mathrm{CD}(s,e)}{M \cdot N^2}$$

where CD represents number of correctly detected songs which can transit emotion state from emotion s to emotion e. M is the number of songs in the test set, and N describes the number of emotions.

Second, STD represents the distribution of measured accuracy values. High STD means that accuracy values are dispersed broadly. This indicates the possibility of a big difference

Table 3 Overall performance of our system with original SVM

Kernel Function	# of folds	Overall Accuracy (%)	STD of Accuracy	Maximum Accuracy (%)
Linear	2	56.34	0.1331	78.33
	3	63.73	0.1378	82.50
	4	66.02	0.1337	81.11
	5	66.99	0.1377	86.46
	6	58.95	0.1323	81.00
Polynomial	2	56.43	0.1323	78.33
	3	60.06	0.1395	85.00
	4	65.79	0.1367	87.78
	5	63.13	0.1384	81.25
	6	56.70	0.1413	81.00
Radial Basis	2	57.25	0.1258	75.00
	3	66.71	0.1208	77.50
	4	66.62	0.1242	80.00
	5	59.60	0.1283	81.25
	6	58.26	0.1243	80.00
Sigmoid	2	55.83	0.1197	75.00
	3	65.89	0.1129	72.50
	4	67.54	0.1151	78.89
	5	62.76	0.1135	78.13
	6	61.57	0.1166	74.00

between maximum and minimum values of the data. If overall accuracy is not good but STD accuracy is high, then it is possible to have higher maximum accuracy and lower minimum accuracy.

Third, maximum accuracy represents the best performance that the classifier can give. If the maximum accuracy is high, then it indicates good performance in a specific data set or specified emotion state transition.

Tables 3 and 4 show the overall performance of original SVM and one-class SVM, respectively. The tables show that performance depends strongly on the SVM, kernel function, and the number of folds in the cross validation. In Table 3, Sigmoid function+ 4-fold cross validation shows the best performance, and both linear kernel and radial basis kernel give next best overall accuracy. Also, 4-fold cross validation showed best result both in overall accuracy (67.54% in original SVM and sigmoid kernel function) and maximum accuracy (87.78% in one-class SVM and polynomial kernel function). Also, the STD of accuracy and maximum accuracy shows a propositional relationship as shown in Fig. 14.

6.3.2 Original SVM vs. one-class SVM

In the comparison of overall accuracy, original SVM is better than one-class SVM. However, in terms of maximum accuracy, the result is opposite. Figures 15(a) and (b) show the comparison of original SVM and one-class SVM in terms of overall accuracy and maximum accuracy, respectively. The dashed line represents the borderline where original SVM is better than one-class SVM, and vice verse. In the figure, black points indicate data

Table 4 Overall performance of our system with one-class SVM

Kernel Function	# of folds	Overall Accuracy (%)	STD of Accuracy	Maximum Accuracy (%)
Linear	2		0.1294	85.00
	3	64.32	0.1365	82.50
	4	63.82	0.1341	83.33
	5	63.59	0.1376	81.25
	6	55.79	0.1324	77.00
Polynomial	2	55.23	0.1406	88.33
	3	62.99	0.1469	86.25
	4	64.05	0.1423	87.78
	5	57.07	0.1442	85.42
	6	61.07	0.1461	85.00
Radial Basis	2	51.65	0.1376	85.00
	3	64.51	0.1373	82.50
	4	64.23	0.1335	82.22
	5	64.97	0.1288	84.38
	6	60.61	0.1349	81.00
Sigmoid	2	54.32	0.1214	81.67
	3	58.95	0.1307	81.25
	4	66.44	0.1260	75.56
	5	61.11	0.1293	83.33
	6	58.40	0.1223	79.00

points where original SVM is better than one-class SVM. These points were generated by combining the data in Tables 3 and 4. For instance, the point at (66.99, 63.59) was generated from the fourth records of Tables 3 and 4. Since the point is below the borderline, it was marked black.

In Figure 15(a), only 5 points are above the borderline. Also, average and maximum differences of overall accuracy between original SVM and one-class SVM are 1.35% and 6.94%, respectively. Thus, we can conclude that original SVM is better than one-class SVM

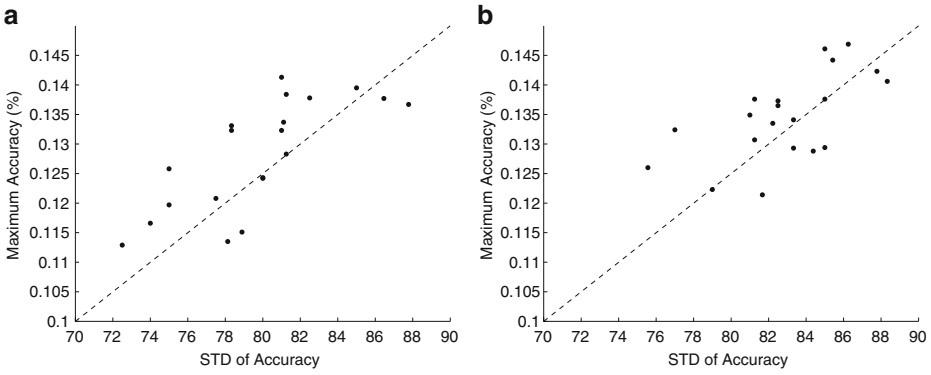


Fig. 14 Relationship between STD of accuracy and maximum accuracy in **a** original SVM, and **b** one-class SVM

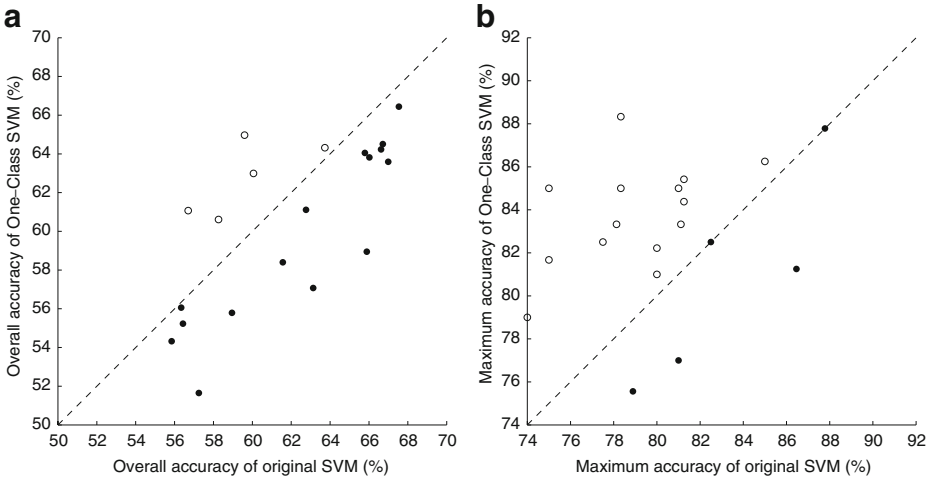


Fig. 15 Comparison of **a** overall accuracy, and **b** maximum accuracy between original SVM and one-class SVM

in terms of average accuracy. However, in Fig. 15(b), one-class SVM was more accurate than original SVM. Only 3 points are in the area of original SVM, and 2 points are on the borderline. Other points show that one-class SVM is better than original SVM. Furthermore, statistical analysis shows that the average and maximum differences of maximum accuracy are 3.14%, and 10.00%, respectively.

The experiment results show that original SVM and one-class SVM have their own advantages and disadvantages. As shown in Fig. 15(a), the overall accuracy of original SVM is better than one-class SVM. Thus, original SVM gives better results than one-class SVM, on the average. However, one-class SVM might give better result in terms of maximum accuracy. As described in Fig. 15(b), one-class SVM won over original SVM in maximum accuracy. This indicates that one-class SVM is better than original SVM in some special cases. This can be used in music information retrieval as follows: if the listener's musical preference or taste is biased toward some specific music, then one-class SVM

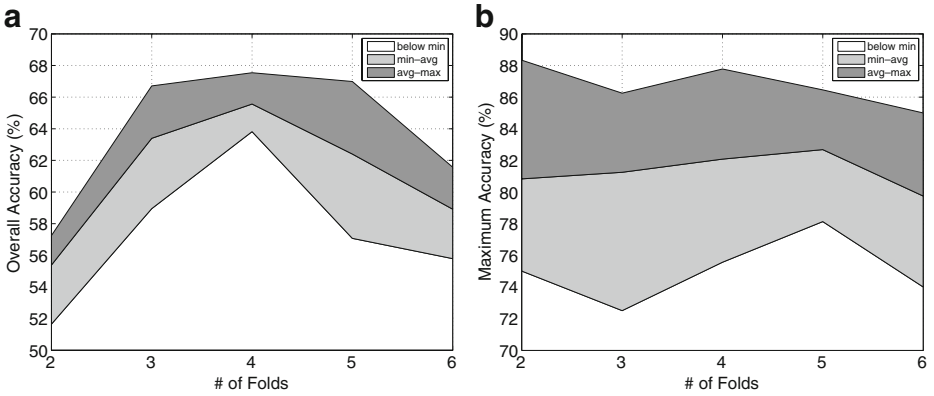


Fig. 16 Minimum, average, and maximum of **a** overall accuracy, and **b** maximum accuracy

might be better for personalized service. Therefore, if listeners' musical tastes need to be grouped or managed in groups, then original SVM might be better.

6.4 Determining the number of folds

The number of folds can also affect the accuracy of SVM prediction. In our experiment, we tested different numbers of folds to empirically find the best one. The size of our dataset is 120, and we used common divisors of 120 to decide the number of folds. This is because by using a common divisor of dataset size, we can avoid any leftovers from the dataset after grouping.

Figure 16 presents the results of *v*-fold cross validation. It shows that 4-fold cross validation is better than the others. Figure 16(a) shows cross-validation for the overall accuracy of all SVMs and kernels. This graph clearly shows that 4-fold cross validation is best for training. The minimum and maximum bounds of 4-fold cross validation are the highest among all results. Figure 16(b) shows that 2-fold or 4-fold are best in terms of maximum accuracy. Both 2-fold and 4-fold are best in maximum-bound.

7 Conclusion

In this paper, we proposed a novel context-aware music recommendation system. Our contributions are as follows: (1) we proposed ESTM to represent a human being's emotion state transition in a mathematical framework. Under this framework, emotions become a measurable and analyzable entity, especially for music recommendation. (2) We proposed COMUS ontology for evaluating the user's desired emotion state based upon the user's situation and preferences. Based on current and desired emotions and ESTM, the system selects appropriate music from the database. (3) To accomplish the aforementioned music recommendation, we proposed a novel music classification based on low level features. In particular, we performed a framing signal sequence using a beat tracking algorithm. For each frame, we extracted various acoustic features including new features such as variation of beat lengths and chord/scale variations. Other features considered in the experiment include AE, SC, SF, SS, SFX, and MFCC. Because of *the curse of dimensionality* problem, we used NMF for dimensionality reduction with low information loss.

We collected various scenarios which describe the user's situation and preferences. We also collected data for user emotion state transitions through a web-based questionnaire. With the support of COMUS ontology and ESTM ground-truth, we could evaluate emotion state transitions from the user's context information. In order to map ESTM from low level music features, we used SVM and carried out various experiments with diverse factors. In our experiments, we achieved 67.54% overall accuracy and 87.78% maximum accuracy.

Acknowledgement This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD). (KRF-2007-313-D00758)

References

1. Allwein E, Schapire R, Singer Y (2000) Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1:113–141
2. Boser BE, Guyon I and Vapnik V (1992) A training algorithm for optimal margin classifiers. In proceedings of the Fifth Annual Workshop on Computational Learning Theory : 144–152, ACM Press

3. Chang C-C and Lin C-J (2001) LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm of subordinate document. Access 30 Jun 2008.
4. Cord M, Cunningham P (2008) Machine Learning Techniques for Multimedia. Springer-Verlag Berlin Heidelberg.
5. Cortes C, Vapnik V (1995) Support-vector network. Machine Learning 20:273–297
6. Ellis D, and Poliner G (2007) Identifying `Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007) 4:1429–1432
7. Feng Yazhong , Zhuang, Yueting, Pan Yunhe. (2003) Popular Music Retrieval by Detecting Mood. SIGIR Forum (ACM Special Interest Group on Information Retrieval), Page(s): 375–376.
8. Grüninger M, Fox MS (1994) The Role of Mariano Fernández López 4–12 Competency Questions in Enterprise Engineering. IFIP WG 5.7 Workshop on Benchmarking. Theory and Practice. Trondheim, Norway.
9. Han B, Hwang E, Rho S, Kim M (2007) M-MUSICS: Mobile Content-based Music Retrieval System. ACM Multimedia 2007:496–497
10. Holzapfel A, Stylianou Y (2008) Musical Genre Classification Using Nonnegative Matrix Factorization-Based Features. IEEE Transactions on Audio, Speech, and Language Processing 16(2):424–434
11. ISO 226:2003 Acoustics — Normal equal-loudness level contours.
12. Jun S, Rho S, Han B, Hwang E (2008) A Fuzzy Inference-based Music Emotion Recognition System. IEEE International Conferences on Visual Information Engineering 2008 (VIE '08), to appear in Jul. 2008
13. Juslin PN (2000) Cue utilization in communication of emotion in music performance: Relating performance to perception. J. Experimental Psychology 26:1797–1813
14. Kalat JW and Shiota MN (2007) Emotion. Thomson. 1/e.
15. Klapuri A(ed.) and Davy, M(ed.) (2006) Signal Processing Methods for Music Transcription. Springer Science + Business Media LLC.
16. Krumhansl C (1990) Cognitive Foundations of Musical Pitch. Oxford University Press.
17. Lee DD, Seung S (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755):788–791
18. Lee DD and Seung S (2001) Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing System 13 : Proceedings of the 2000 Conference. 556–562. MIT Press.
19. Lu L, Liu D, Zhang H-J (2006) Automatic MoodDetection and Tracking of Music Audio Signals. IEEE Transactions on Audio, Speech, and Language Processing 14(1):5–18
20. Müller M (2008) Information Retrieval for Music and Motion. Springer Berlin Heidelberg New York
21. Oscar C (2006) Foafing the Music: Bridging the semantic gap in music recommendation. Proceedings of 5th International Semantic Web Conference (ISWC)
22. Oscar C, Perfecto H, and Xavier S (2006) A multimodal approach to bridge the Music Semantic Gap. International Conference on Semantic and Digital Media Technologies (SAMT)
23. Pachet F, Casaly D (2000) "A taxonomy of musical genres," in Proceedings of the 6th Conference on Content-Based Multimedia Information Access (RIAO'00). France, April, Paris
24. Rabiner LR, Juang B-H (1993) Fundamentals of Speech Recognition. Prentice Hall, New Jersey
25. Rho S, Han B, Hwang E, Kim M (2008) MUSEMBLE: A Novel Music Retrieval System with Automatic Voice Query Transcription and Reformulation. Journal of Systems and Software 81(7):1065–1080
26. Robinson DW et al (1956) A re-determination of the equal-loudness relations for pure tones. British Journal of Applied Physics 7:166–181
27. Russel JA (1980) A circumplex model of affect. Journal of Personality Social Psychology 39:1161–1178
28. Scherer K (1992) What does facial expression express? In K.T. Strongman(Ed.) International Review of Studies on Emotions 2:139–165. Chichester: Wiley.
29. Scherer KR (2005) What are emotions? And how can they be measures? Social Science Information 44 (4):695–729
30. Schölkopf B, Platt JC et al (1999) Estimating the support of a high-dimensional distribution. Microsoft research corporation technical report MSR-TR-99-87.
31. Thayer RE (1989) The Biopsychology of Mood and Arousal. Oxford Univ. Press, Oxford, U.K
32. Xiang H, Ren F, Kuroiwa S, Jiang P (2005) An Experimentation on Creating a Mental State Transition Network. Proceedings of the 2005 IEEE International Confernece on Information Acquisition:432–436
33. Yazhong Feng; Yueting Zhuang; Yunhe Pan. (2003) Music information retrieval by detecting mood via computational media aesthetics. Web Intelligence, Proceedings. IEEE/WIC International Conference onVolume, Issue, 13–17. Page(s): 235 – 241.
34. Yves R and Frederick G (2007) Music Ontology Specification, http://www.musicontology.com/
35. Yves R, Samer A, Mark S, Frederick G (2007) The Music Ontology. Proceedings of the International Conference on Music Information Retrieval, ISMIR 2007:417–422

36. Zwicker E, Fastl H (1990) Psychoacoustics — Facts and Models. (1st Ed.) Springer.
37. Kanzaki Music Vocabulary. http://www.kanzaki.com/ns/music
38. MusicBrainz. http://musicbrainz.org
39. Music Ontology Specification, http://www.musicontology.com/
40. COMUS Ontology, http://mil.korea.ac.kr/ontology/0.1/musicontology.owl
41. All Music Guide, http://www.allmusic.com/
42. BPM Database, http://www.bpmdatabase.com/
43. Protégé editor, http://protege.stanford.edu/
44. Mood Logic, Available at: http://www.moodlogic.com/

Byeong-jun Han received his B.S. and M.S. degrees in electrical engineering from Korea University, Korea, in 2005 and 2007, respectively. Currently he is pursuing the Ph.D. degree in the School of Electrical Engineering in Korea University. He is currently working on audio analysis and intelligent music information retrieval (MIR) system development. Mr. Han has been a reviewer in Multimedia Tools and Applications (MTAP) and Information Science (Elsevier). His research interests MIR, environmental sound classification, acoustic analysis and classification, multimedia feature extraction, audio/visual retrieval system, multimedia data mining, and machine learning.

Seungmin Rho received his M.S. and Ph.D. degrees in Computer Science from Ajou University, Korea, in 2003 and 2008, respectively. With research fellowships, he is currently working as a postdoctoral fellow at Prof. Roger Dannenberg's research group at the Department of Computer Science and Music Lab in Carnegie Mellon University. Dr. Rho's research interests include database, music retrieval, multimedia systems, machine learning, knowledge management and intelligent agent technologies. Dr. Rho has been a reviewer in Multimedia Tools and Applications (MTAP), Journal of Systems and Software, Information Science (Elsevier), and Program Committee member in over 10 international conferences. He has published 10 articles in journals and book chapters and 18 in international conferences and workshops. He is listed in Who's Who in the World.

Sanghoon Jun received his B.S. degree in Electrical Engineering from Korea University, Korea, in 2008. Currently he is pursuing the M.S. degree in the School of Electrical Engineering in Korea University. He is currently working on audio analysis and intelligent music retrieval system development. His research interests include multimedia recommendation, content based music retrieval/recommendation, semantic multimedia and machine learning.

Eenjun Hwang received his B.S. and M.S. degree in Computer Engineering from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively; and his Ph.D. degree in Computer Science from the University of Maryland, College Park, in 1998. From September 1999 to August 2004, he was with the Graduate School of Information and Communication, Ajou University, Suwon, Korea. Currently he is a member of the faculty in the School of Electrical Engineering, Korea University, Seoul, Korea. His current research interests include database, multimedia systems, audio/visual feature extraction and indexing, semantic multimedia, information retrieval and Web applications.